Attorney's Docket No.: 06975-206001 / Network 08

OIPE IAP38 FEB 2 1 2006 PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Patrick Meenan
Serial No. : 09/867,546
Filed : May 31, 2001
Title : LOCAL PROTOCOL SERVER

Art Unit : 2666
Examiner : Michael J. Moore, Jr.

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL LETTER AND PETITION FOR AUTOMATIC EXTENSION

Correspondence relating to this application is enclosed. The required fees are computed below. Please apply any charges, or any credits, to Deposit Account No. 06-1050.

Appeal Brief fee	$500
Applicant hereby petitions under 37 C.F.R. § 1.136 for a 1 month extention of time	$120
TOTAL FEE DUE	$620

Respectfully submitted,

Date: 2/21/2006

W. Karl Renner
Reg. No. 41,265

Fish & Richardson P.C.
Citigroup Center
52nd Floor
153 East 53rd Street
New York, New York 10022-4611
Telephone: (212) 765-5070
Facsimile: (212) 258-2291

02/23/2006 HDAHTE1 00000002 061050 09867546
02 FC:1251 120.00 DA

30269061.doc

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Patrick Meenan
Serial No. : 09/867,546
Filed : May 31, 2001
Title : LOCAL PROTOCOL SERVER

Art Unit : 2666
Examiner : Michael J. Moore, Jr.

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

(1) Real Party in Interest

America Online, Inc. is the real party in interest.

(2) Related Appeals and Interferences

There are no related appeals or interferences.

(3) Status of Claims

Claims 1-64 are pending, with claims 1, 14, 28, and 42 being independent.

(4) Status of Amendments

All amendments have been entered.

(5) Summary of Claimed Subject Matter

Independent claim 1 is directed to a system located on a client device for communicating data packets between the client device and a host system. See, e.g., Application, Page 9, Line 19 to Page 12, Line 31; Figs. 4-11. The system comprises a protocol server module. See, e.g., Application, Page 10, Lines 12-17; Fig. 4, Element 477. The protocol server module terminates a communication session that uses a first protocol and that is intended to enable communications between a source and a destination, wherein the source is one of the client device operating system protocol stack and the host system and the destination is one of the client device operating system protocol stack and the host system but differs from the source. See, e.g., Application, Page 10, Lines 18-30; Fig. 4, Elements 475, 477. The protocol server module translates data packets from the source between the first

protocol and a second protocol that differs from the first protocol. See, e.g., Application, Page 12, Lines 3-9. The protocol server module receives configuration data assigned by the host system, and transports the data packets having the second protocol to the destination using the configuration data assigned by the host system. See, e.g., Application, Page 15, Lines 16-22; Fig. 5, Elements 540, 550. In addition, the system comprises a controller module that is logically connected to the protocol server module. See, e.g., Application, Page 15, Lines 15-21; Fig. 4, Element 479. The controller module controls communications between the client device operating system protocol stack, the protocol server module, and the host system. See, e.g., Application, Page 15, Lines 15-21.

Independent claim 14 is a method claim that includes several limitations corresponding to those recited in claim 1 and that is therefore supported by many of the portions of the specification and figures identified above with respect to claim 1.

Independent claim 28 is directed to a system located on a client device for communicating data packets between the client device and a host system. See, e.g., Application, Page 9, Line 19 to Page 12, Line 31; Figs. 4-11. The system comprises a protocol server module . See, e.g., Application, Page 10, Lines 12-17; Fig. 4, Element 477. The protocol server module terminates a communication session between a source and a destination, wherein the source is one of a client device operating system protocol stack and the host system and the destination is one of the client device operating system protocol stack and the host system but differs from the source. See, e.g., Application, Page 10, Lines 18-30; Fig. 4, Elements 475, 477. In addition, the protocol server module receives configuration data assigned by the host system. See, e.g., Application, Page 15, Lines 16-22; Fig. 5, Elements 540, 550. Finally, the protocol server module transports data packets to the destination through a network using any one of several communication protocols using the configuration data assigned by the host system. See, e.g., Application, Page 15, Lines 16-22. The system also comprises a controller module that is logically connected to the protocol server module. See, e.g., Application, Page 15, Lines 15-21; Fig. 4, Element 479. The controller module controls communications between the client device operating system protocol stack, the protocol server module, and the host system. See, e.g., Application, Page 15, Lines 15-21.

Independent claim 42 is a method claim that includes several limitations corresponding to those recited in claim 14 and that is therefore supported by many of the portions of the specification and figures identified above with respect to claim 14.

(6) Grounds of Rejection

Claims 1-64 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Chiles et al. (U.S. 6,618,393) in view of Ra et al. (U.S. 6,577,643).

(7) Argument

The Rejection Of Claims 1-64 as being obvious over Chiles in view of Rai should be reversed.

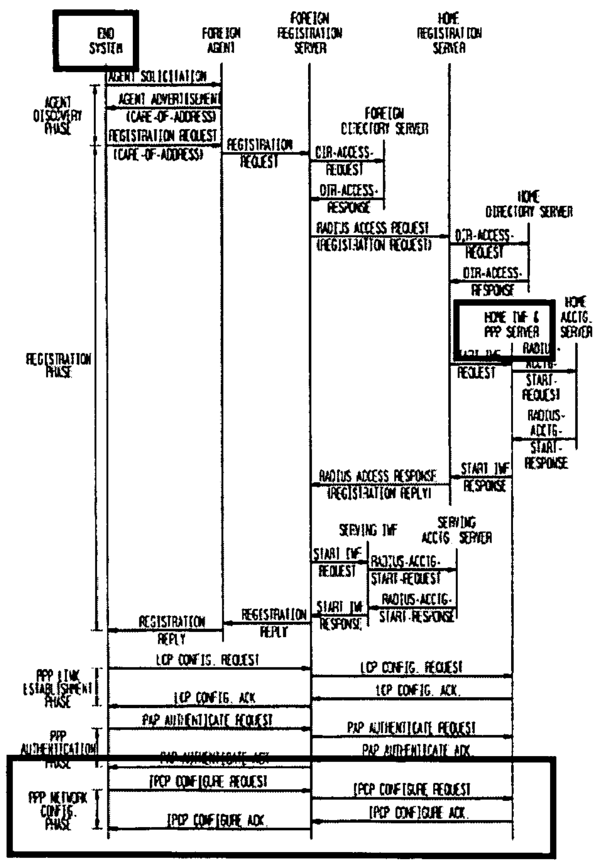
The combination of Chiles and Rai fails to disclose a protocol server receiving "configuration data assigned by the host system", and thus fails to support the presently pending rejection of claims 1, 14, 28, and 42 under 35 U.S.C. §103(a).

Independent claim 1 recites a protocol server structured and arranged to "receive configuration data assigned by the host system." Chiles fails to teach this limitation. Instead of receiving configuration data from the host system, Chiles explicitly teaches its readers to obtain configuration data from a local user. <u>See</u> elem. 56, Fig. 2 ("User Provided Address Configuration (CIP, IPX, Etc)"). In fact, the Final Office Action directly acknowledges that Chiles fails to meet this limitation. <u>See</u> Final Office Action from August 3, 2005, pages 3-4 ("Chiles does not explicitly teach the reception of configuration data by the host system and the transporting of translated data packets to the destination using this configuration data.").

Recognizing the shortcomings of Chiles, the Office Action appears to rely upon Rai to suggest the obviousness of providing the spoofing module (protocol server) of Chiles with a host-assigned configuration data. Specifically, the Office Action contends that Rai shows a PPP server assigning an IP address to a client device via IP Control Protocol (IPCP), argues that modifying Chiles to integrate this functionality would have been obvious, and contends that such a modification would suggest the claimed provision of host-supplied configuration data to the protocol server. <u>See</u> Final Office Action from August 3, 2005, page 4.

Applicant disagrees. The combination of Chiles and Rai is believed to be redundant and to continue to fail to disclose the claimed limitation. Specifically, the functionality taught by Rai is already disclosed in Chiles – with neither of Rai nor Chiles suggesting the claimed limitation of providing host-supplied configuration data to a protocol server.

FIG. 14



Rai teaches a PPP server that assigns an IP address to a PPP client via IPCP. See Rai, Col. 10:11-25; Col. 19:4-6; see also Fig. 14 (reproduced herein). As shown in the relevant part of Fig. 14 on the right, the client (end system) issues an IPCP Configure Request to the PPP server (Home IWF & PPP Server). The PPP assigns an IP address to the client and transmits it to the client via IPCP Configure ACK. Therefore, Rai merely teaches the PPP server assigning the IP address to the client, not the PPP server obtaining the IP address from another host. Rai does not describe how the PPP server obtains the IP address before assigning it to the client.

This teaching of Rai is similar to Chiles, which also discloses a PPP peer emulation module 52 (protocol server) negotiating an IP address with a PPP module (client) via IPCP. See, e.g., Col. 6: 43-48. Yet, each of Chiles and Rai fail to disclose the provision of host-supplied configuration address information to the PPP server.

In contrast to both Chiles and Rai, the rejected claims require that a protocol server receive configuration data, such as IP address, from the host system. In other words, the claimed limitation describes how the protocol server obtains configuration data, but is not concerned with how the protocol server later assigns this data to a client. Chiles, Rai, or their combination do not teach this specific limitation. By contrast, Chiles is clear in its indication to the contrary – expressly indicating that the PPP peer emulation module (protocol server)

obtains the IP address (configuration data) from the user. And Rai is altogether silent on how the PPP server obtains the IP address before assigning it to the client.

As a result, Chiles and Rai fail, expressly and implicitly, alone or in combination, to disclose or suggest the claimed protocol server receiving configuration data assigned by the host system. Similar to claim 1, the independent claims 14, 28, and 42 also recite receiving "configuration data assigned by the host system" and therefore should be allowable for the reasons articulated for claim 1.

With respect to claims 2-13, 15-27, 29-41, and 43-64, Applicants respectfully request reconsideration and withdrawal of their rejection under 35 U.S.C. §103 because Chiles and Rai, either alone or in combination, fail to describe or suggest the features discussed above with respect to independent claims 1, 14, 28, and 42 upon which claims 2-13, 15-27, 29-41, and 43-64 depend.

The brief fee of $500 is enclosed. Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: 2/21/2006

W. Karl Renner
Reg. No. 41,265

Fish & Richardson P.C.
Citigroup Center
52nd Floor
153 East 53rd Street
New York, New York 10022-4611
Telephone: (212) 765-5070
Facsimile: (212) 258-2291

30268410.doc

(8) Appendix of Claims

1. (Rejected) A system located on a client device for communicating data packets between the client device and a host system, the system comprising:

a protocol server module structured and arranged to:

terminate a communication session that uses a first protocol and that is intended to enable communications between a source and a destination, wherein the source is one of the client device operating system protocol stack and the host system and the destination is one of the client device operating system protocol stack and the host system but differs from the source,

translate data packets from the source between the first protocol and a second protocol that differs from the first protocol,

receive configuration data assigned by the host system, and

transport the data packets having the second protocol to the destination using the configuration data assigned by the host system; and

a controller module that is logically connected to the protocol server module and that is structured and arranged to control communications between the client device operating system protocol stack, the protocol server module, and the host system.

2. (Rejected) The system of claim 1 wherein the data packets include an encapsulation and the protocol server module translates the data packets by removing the encapsulation from the data packets.

3. (Rejected) The system of claim 1 wherein the protocol server module translates the data packets by encapsulating the data packets using any one of several communication protocols that differs from the first protocol.

4. (Rejected) The system of claim 2 wherein the protocol server module translates the data packets to further include encapsulating the data packets using any one of several communication protocols that differs from the first protocol.

5. (Rejected) The system of claim 1 wherein the client device operating system protocol stack supports PPP.

6. (Rejected) The system of claim 5 wherein the protocol server module includes a PPP server module located on the client device.

7. (Rejected) The system of claim 6 wherein the PPP server module is structured and arranged to terminate a PPP communication session between the client device operating system protocol stack and the host system.

8. (Rejected) The system of claim 6 wherein the PPP server module is structured and arranged to negotiate a PPP communication session with the client device operating system protocol stack.

9. (Rejected) The system of claim 1 wherein the protocol server module and the controller module perform transparent to a sender of the data packets.

10. (Rejected) The system of claim 1 wherein the protocol server module is structured and arranged to enable collection of data for error checking.

11. (Rejected) The system of claim 1 wherein the protocol server module is structured and arranged to filter the data packets prior to transporting the data packets to the destination.

12. (Rejected) The system of claim 1 further comprising a virtual modem adapter logically connected between the client device operating system protocol stack and the protocol server module.

13. (Rejected) The system of claim 1 wherein the data packets include layer three data packets.

14. (Rejected) A method for communicating data packets between a client device and a host system through a network, the method comprising:

at the client device, terminating a communication session that uses a first protocol and that is intended to enable communications between a source and a destination, wherein the source is one of a client device operating system protocol stack and the host system and the destination is one of the client device operating system protocol stack and the host system but differs from the source;

translating data packets from the source between the first protocol and a second protocol that differs from the first protocol;

receiving configuration data assigned by the host system; and

transporting the data packets having the second protocol to the destination using the configuration data assigned by the host system.

15. (Rejected) The method as in claim 14 wherein the data packets include an encapsulation and translating the data packets includes removing the encapsulation from the data packets.

16. (Rejected) The method as in claim 14 wherein translating the data packets includes encapsulating the data packets using any one of several communication protocols that differs from the first protocol.

17. (Rejected) The method as in claim 15 wherein translating the data packets further includes encapsulating the data packets using any one of several communication protocols that differs from the first protocol.

18. (Rejected) The method as in claim 14 wherein:

terminating the communication session includes using a protocol server module to terminate the communication session;

translating the data packets includes using the protocol server module to translate the data packets from the source between the first protocol and the second protocol; and

transporting the data packets includes using the protocol server module to transport the data packets having the second protocol to the destination.

19. (Rejected) The method as in claim 18 wherein the client device operating system protocol stack supports PPP.

20. (Rejected) The method as in claim 19 wherein the protocol server module includes a PPP server module located on the client device.

21. (Rejected) The method as in claim 20 wherein terminating the communication session includes using the PPP server module to terminate a PPP communication session between the client device operating system protocol stack and the host system.

22. (Rejected) The method as in claim 20 wherein terminating the communication session includes using the PPP server module to negotiate a PPP communication session with the client device operating system protocol stack.

23. (Rejected) The method as in claim 18 further comprising using a virtual modem adapter to interface between the client device operating system protocol stack and the protocol server module.

24. (Rejected) The method as in claim 14 wherein terminating the communication session, translating the data packets, and transporting the data packets occur transparently to a sender of the data packets.

25. (Rejected) The method as in claim 14 further comprising enabling collection of data for error checking.

26. (Rejected) The method as in claim 14 further comprising filtering the data packets prior to transporting the data packets to the destination.

27. (Rejected) The method as in claim 14 wherein the data packets include layer three data packets.

28. (Rejected) A system located on a client device for communicating data packets between the client device and a host system, the system comprising:

a protocol server module structured and arranged to:

terminate a communication session between a source and a destination, wherein the source is one of a client device operating system protocol stack and the host system and the destination is one of the client device operating system protocol stack and the host system but differs from the source,

receive configuration data assigned by the host system, and

transport data packets to the destination through a network using any one of several communication protocols using the configuration data assigned by the host system; and

a controller module that is logically connected to the protocol server module and that is structured and arranged to control communications between the client device operating system protocol stack, the protocol server module, and the host system.

29. (Rejected) The system of claim 28 wherein the protocol server module is structured and arranged to translate the data packets prior to transporting the data packets.

30. (Rejected) The system of claim 29 wherein the data packets include an encapsulation and the protocol server module translates the data packets by removing the encapsulation from the data packets.

31. (Rejected) The system of claim 29 wherein the protocol server module translates the data packets by encapsulating the data packets using any one of several communication protocols that differs from an original protocol.

32. (Rejected) The system of claim 30 wherein the protocol server module translates the data packets to further include encapsulating the data packets using any one of several communication protocols that differs from an original protocol.

33. (Rejected) The system of claim 28 wherein the client device operating system protocol stack supports PPP.

34. (Rejected) The system of claim 33 wherein the protocol server module includes a PPP server module located on the client device.

35. (Rejected) The system of claim 34 wherein the PPP server module is structured and arranged to terminate a PPP communication session between the client device operating system protocol stack and the host system.

36. (Rejected) The system of claim 34 wherein the PPP server module is structured and arranged to negotiate a PPP communication session with the client device operating system protocol stack.

37. (Rejected) The system of claim 28 wherein the protocol server module and the controller module perform transparent to a sender of the data packets.

38. (Rejected) The system of claim 28 wherein the protocol server module is structured and arranged to enable collection of data for error checking.

39. (Rejected) The system of claim 28 wherein the protocol server module is structured and arranged to filter the data packets prior to transporting the data packets to the destination.

40. (Rejected) The system of claim 28 further comprising a virtual modem adapter logically connected between the client device operating system protocol stack and the protocol server module.

41. (Rejected) The system of claim 28 wherein the data packets include layer three data packets.

42. (Rejected) A method for communicating data packets between a client device and a host system through a network, the method comprising:

at the client device, terminating a communication session between a source and a destination, wherein the source is one of a client device operating system protocol stack and the host system and the destination is one of the client device operating system protocol stack and the host system but differs from the source;

receiving configuration data assigned by the host system; and

transporting data packets to the destination through the network using any one of several communication protocols using the configuration data assigned by the host system.

43. (Rejected) The method as in claim 42 further comprising translating the data packets prior to transporting the data packets.

44. (Rejected) The method as in claim 43 wherein the data packets include an encapsulation and translating the data packets includes removing the encapsulation from the data packets.

45. (Rejected) The method as in claim 43 wherein translating the data packets includes encapsulating the data packets using any one of several communication protocols that differs from an original protocol.

46. (Rejected) The method as in claim 44 wherein translating the data packets further includes encapsulating the data packets using any one of several communication protocols that differs from an original protocol.

47. (Rejected) The method as in claim 42 wherein:

terminating the communication session includes using a protocol server module to terminate the communication session; and

transporting the data packets includes using the protocol server module to transport the data packets to the destination through the network using any one of several communication protocols.

48. (Rejected) The method as in claim 47 wherein the client device operating system protocol stack supports PPP.

49. (Rejected) The method as in claim 48 wherein the protocol server module includes a PPP server module located on the client device.

50. (Rejected) The method as in claim 49 wherein terminating the communication session includes using the PPP server module to terminate a PPP communication session between the client device operating system protocol stack and the host system.

51. (Rejected) The method as in claim 49 wherein terminating the communication session includes using the PPP server module to negotiate a PPP communication session with the client device operating system protocol stack.

52. (Rejected) The method as in claim 47 further comprising using a virtual modem adapter to interface between the client device operating system protocol stack and the protocol server module.

53. (Rejected) The method as in claim 42 wherein terminating the communication session and transporting the data packets occur transparently to a sender of the data packets.

54. (Rejected) The method as in claim 42 further comprising enabling collection of data for error checking.

55. (Rejected) The method as in claim 42 further comprising filtering the data packets prior to transporting the data packets to the destination.

56. (Rejected) The method as in claim 42 wherein the data packets include layer three data packets.

57. (Rejected) The system of claim 1 wherein the configuration data includes IP configuration data.

58. (Rejected) The system of claim 1 wherein the configuration data is received without manual user intervention.

59. (Rejected) The method as in claim 14 wherein the configuration data includes IP configuration data.

60. (Rejected) The method as in claim 14 wherein the configuration data receiving the configuration data includes receiving the configuration data without manual user intervention.

61. (Rejected) The system of claim 28 wherein the configuration data includes IP configuration data.

62. (Rejected) The system of claim 28 wherein the configuration data is received without manual user intervention.

62. (Rejected) The method as in claim 42 wherein the configuration data includes IP configuration data.

64. (Rejected) The method as in claim 42 wherein receiving the configuration data includes receiving the configuration data without manual user intervention.

(9) Evidence Appendix

None

(10) Related Proceedings Appendix

None